

# Python workshop

Week 1: Writing your first program

[barbera@van-schaik.org](mailto:barbera@van-schaik.org)



# About me

- During the day I'm a bioinformatician
- In my spare time I ...
  - Go to concerts and festivals
  - Cook (all cuisines)
  - Read (fantasy, popular science/philosophy, Dutch literature)
  - Make things (sewing, electronics, laser cutting, welding, 3d printing)
  - Look into self-hosted cloud services
  - Grow vegetables in my garden



# Overview of this workshop series

- Week 1: Writing your first program
- Week 2: Make choices and reuse code
- Week 3: Loops and strings
- Week 4: Files and lists
- Week 5: Dictionaries and tuples

# Acknowledgments

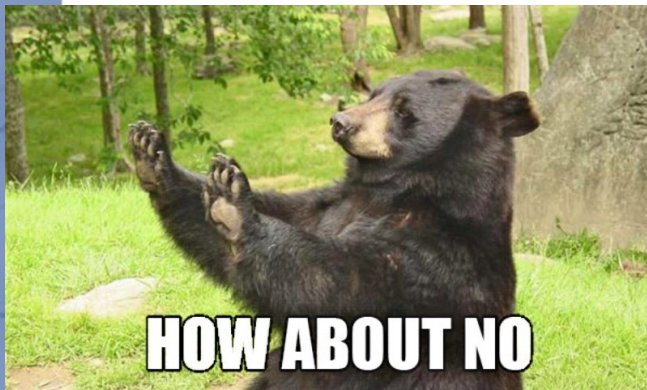
- Charles Severance aka Dr Chuck
  - For teaching me Python
  - For releasing teaching material to the public
  - <http://www.pythonlearn.com/>
- He gives the course “Programming for everybody”
- Structure of this workshop is based on his book
- Several examples too



# Why shouldn't you learn programming?

- Years of learning (to program really well)
- “Coding is not a goal. It's a tool for solving problems.”
- Programming doesn't make you rich
- There are always people better than you

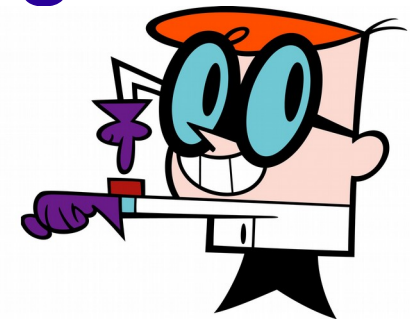
<http://www.fastcolabs.com/3020126/no-you-dont-need-to-learn-to-code>





# Why should you learn programming?

- It is fun and creative
- Because you can
- It is a tool for solving problems



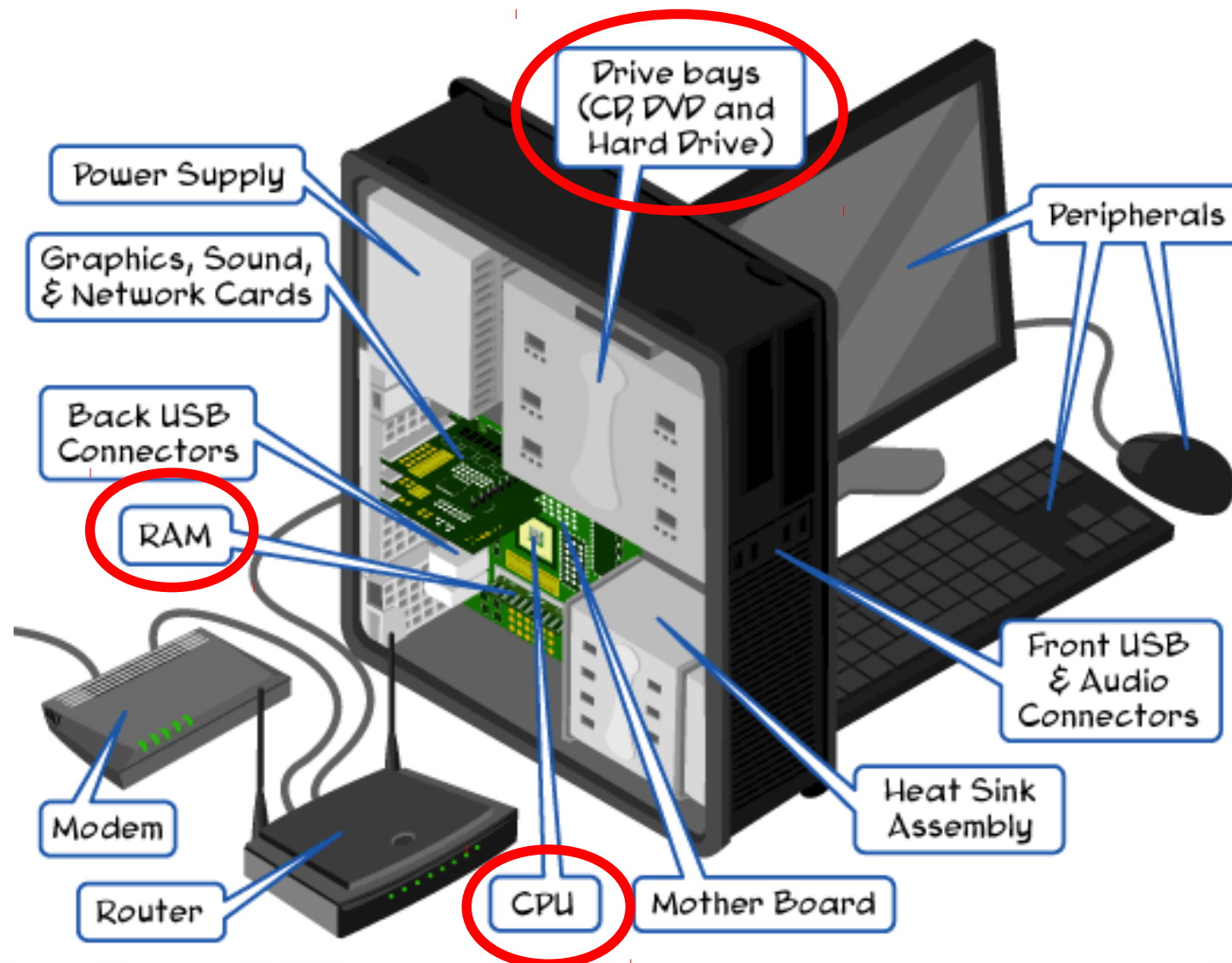
- Science
- Apps
- Games
- Internet of things
- Etc



- A computer is fast and good for repetitive work
- There is no existing program for your needs

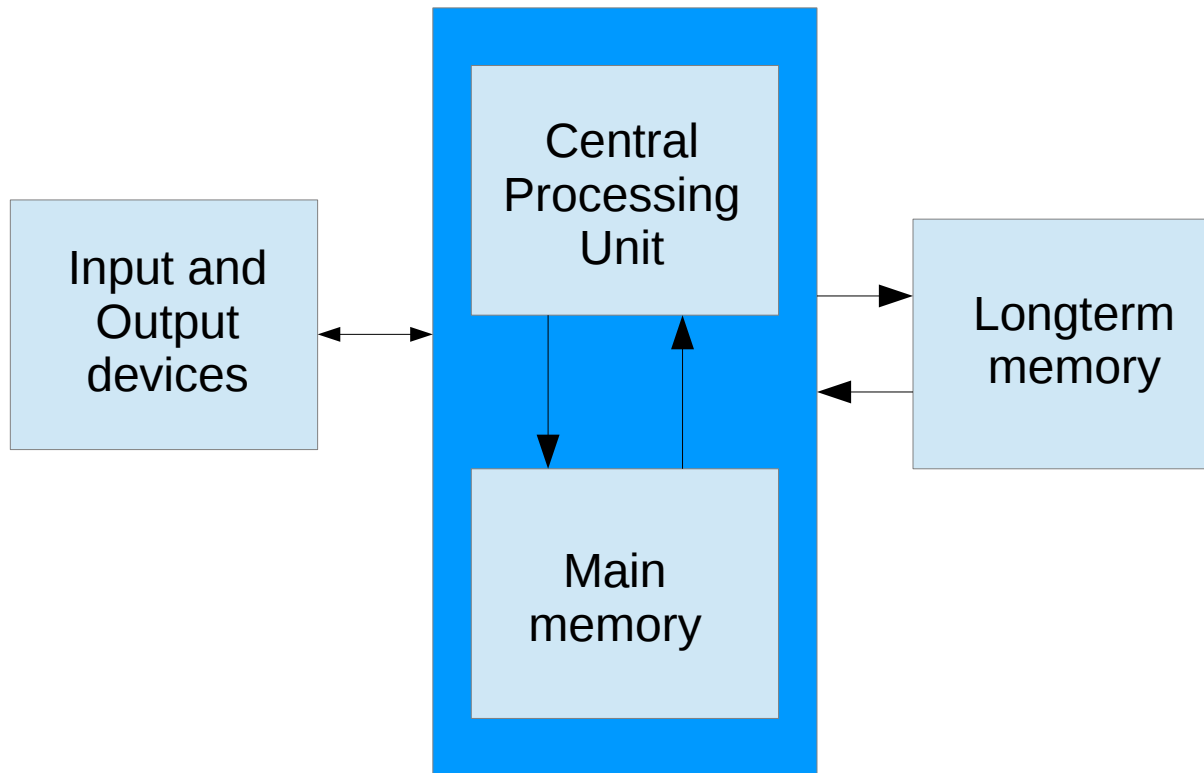
# Computers and language

# Computer anatomy





# Computer anatomy



# Programming language

- To give instructions



<https://youtu.be/XiBYM6g8Tck>

# Programming language

While music is playing:

Left hand out and up

Right hand out and up

Flip left hand

Flip right hand

Left hand to right shoulder

Right hand to right shoulder

Left hand to back of head

Right **ham** to back of head

Left hand to right **hik**

Right hand to left **hik**

Left hand on left bottom

Right hand on right bottom

Wiggle

Wiggle

Jump



# Computer language

```
00101110100110010
10110000001111010
10111010100100101
00001110101111111
01010101001110101
01001010101001010
00100010110011010
01001110100011011
00101110100011000
11101010011101111
```

```
handle = open(filename, 'r')
```

```
song = handle.read()
```

```
words = song.split()
```

```
counts = dict()
```

```
for word in words:
```

```
    counts[word] = counts.get(word, 0) + 1
```

```
for word, count in counts.items():
```

```
    print(word, count)
```

# Why Python

- Easy to learn
- Very readable
  - No long constructs
  - No special cryptic characters
  - Indentation is forced (more next week)
- Much code available (e.g. SciPy, NumPy)
- Cross-platform



<http://helloworldcollection.de/>



# How smart is a computer?

- A computer (or phone) is not smart
  - Limited vocabulary
  - It needs literal instructions
- It will “complain” when instructions are not clear

out of memory

out of disk space

syntax error



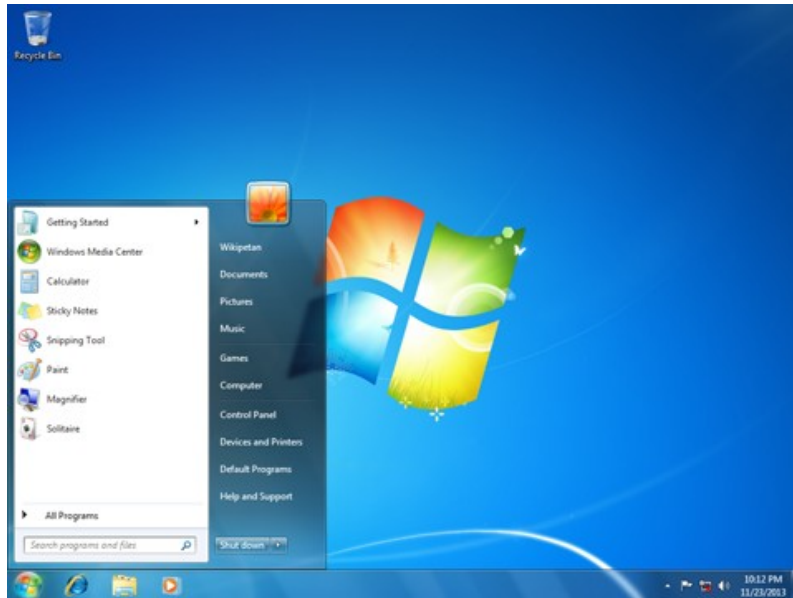
Usually with a hint to guide you

# Reserved words

and as del for is raise  
assert elif from lambda return  
break else global not try  
class except if or while nonlocal  
continue import pass yield  
def finally in with True False  
None



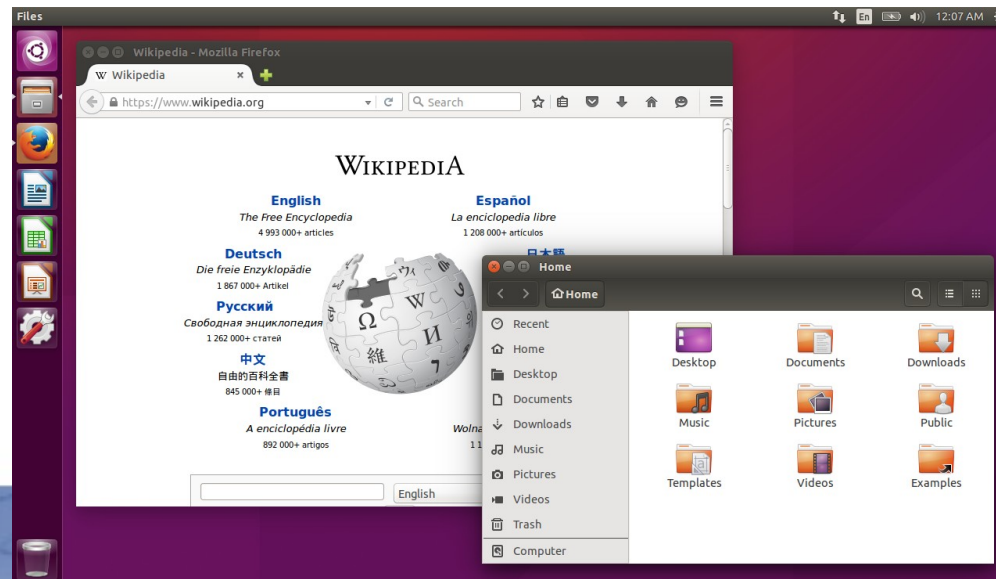
# Terminal, scripts, Python?



Windows



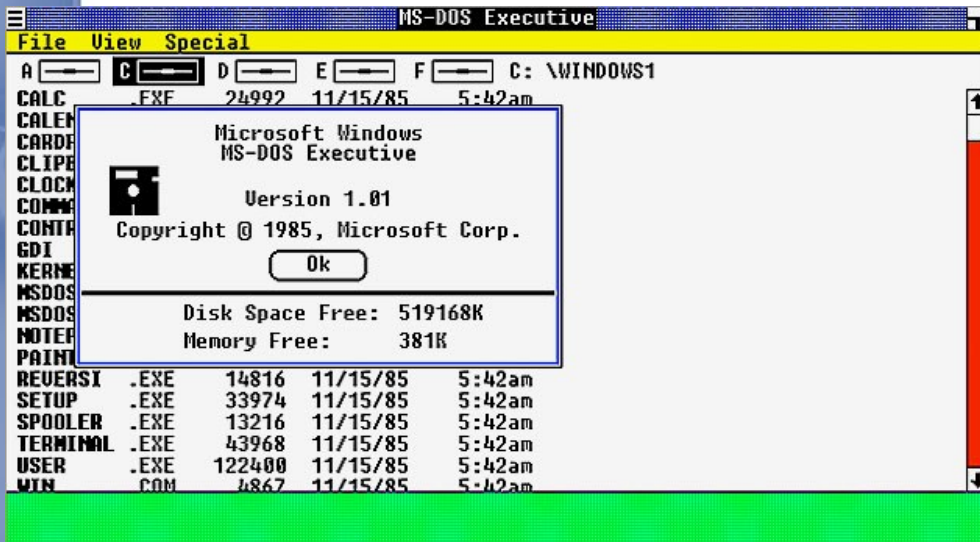
Mac OSX



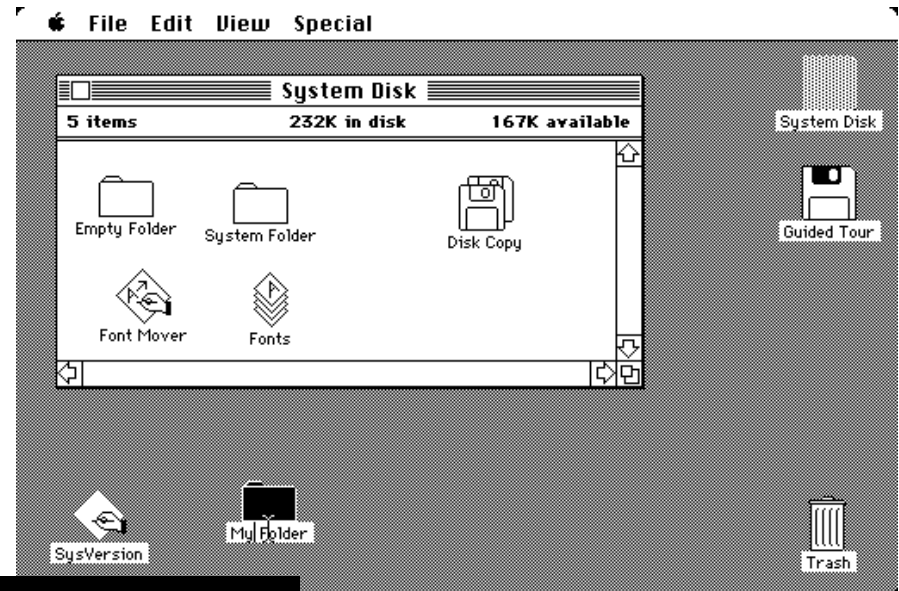
Linux (Ubuntu)



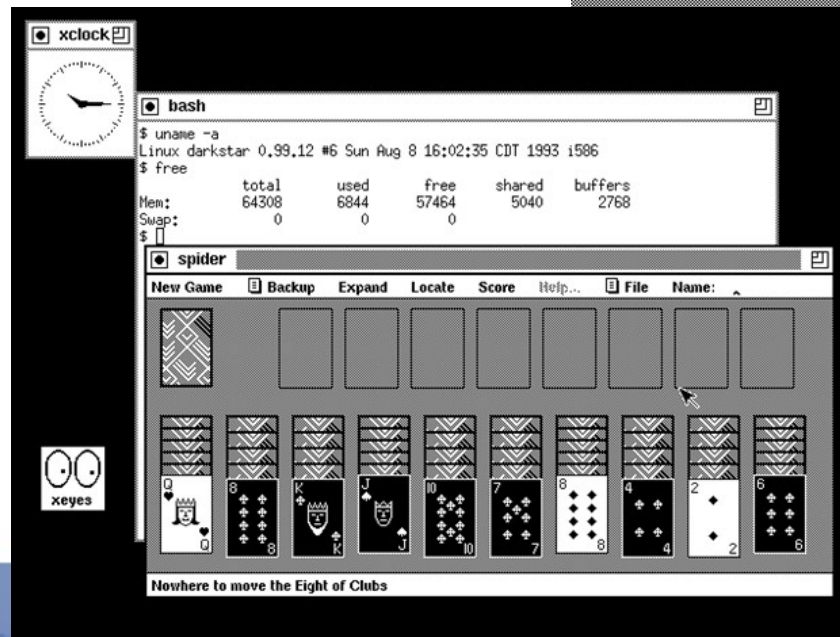
# Vroegah...



Windows



Mac



Linux

# Even earlier: the black window

This is called a:

- Terminal
- Command-line (cmd)
- Shell

```
A:\>dir /w
```

```
Volume in drive A has no label
Volume Serial Number is B83C-90D0
Directory of A:\
```

```

AM2100.DO_      AULCTRA.TXT      COMDEV.IN_      DEPCA.DO_      E20ND.DO_
E21ND.DO_      ELNK.DO_      ELNK16.DO_     ELNK3.DO_      ELNK11.DO_
ELNKM.C.DO_     ELNKPL.DO_     EXP16.DO_      EXPAND.EXE      I82593.DO_
IBMTOK.DO_      IFSHLP.SY_     LICENSE.TXT     LM21DRU.UP_    MSDLC.EX_
NDIS3XR.DO_     NDISHLP.SY_    NET1000.DO_    NE2000.DO_     NET.EX_
NET.MS_          NETBIND.COM     NETH.MS_       N16510.DO_     NWLINK.EXE
OEMDLC.INF      OEMODI.IN_     OEMRAS.IN_     OEMTCPIP.INF   OLITOK.DO_
PE2NDIS.DO_     PROD.IS.DO_     PR04.DO_       PRORAMP.DU_    ROTAND.DO_
PROTHAN.EX_     RASCOPY.BA_     README.TXT     SETUP.EXE_     TCKM.DU_
SMC_ARC.DO_     STRN.DO_        TLNK.DO_       WCNET.INF_     WSETUP.INF
WCSYS.INI       WORKGRP.SY_

```

```
52 file(s)      1,130,630 byte
                  312,822 bytes free
```

```
A:\>
A:\>
A:\>
A:\>
```

Terminal — bash — 67x20

```
bash-3.2$ ls -a
```

```
Users
/home
cores
dev
etc
home
mach_kernel
net
opt
private
sbin
top
usr
var
```

```
User Guides And Information
bash-3.2$ echo $SHELL
/bin/bash
bash-3.2$
```

```

nucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID://
webserverd:x:80:80:WebServer Reserved UID://
postgres:x:90:90:PostgreSQL Reserved UID://:/usr/bin/pfksh
svctag:x:95:12:Service Tag UID://
nobody:x:60001:60001:NFS Anonymous Access User://
noaccess:x:60002:60002:No Access User://
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User://

```

```
"/etc/passwd" 17 lines, 677 characters
```

# ^D

```
testimage console login: root
```

Password:

```
Mar 29 11:36:16 testimage login: ROOT LOGIN /dev/console
```

Last login: Sat Mar 29 11:04:43 on console

Sun Microsystems Inc. SunOS 5.10 Generic January 2005

```
-bash-3.00#
```

## No fancy windows

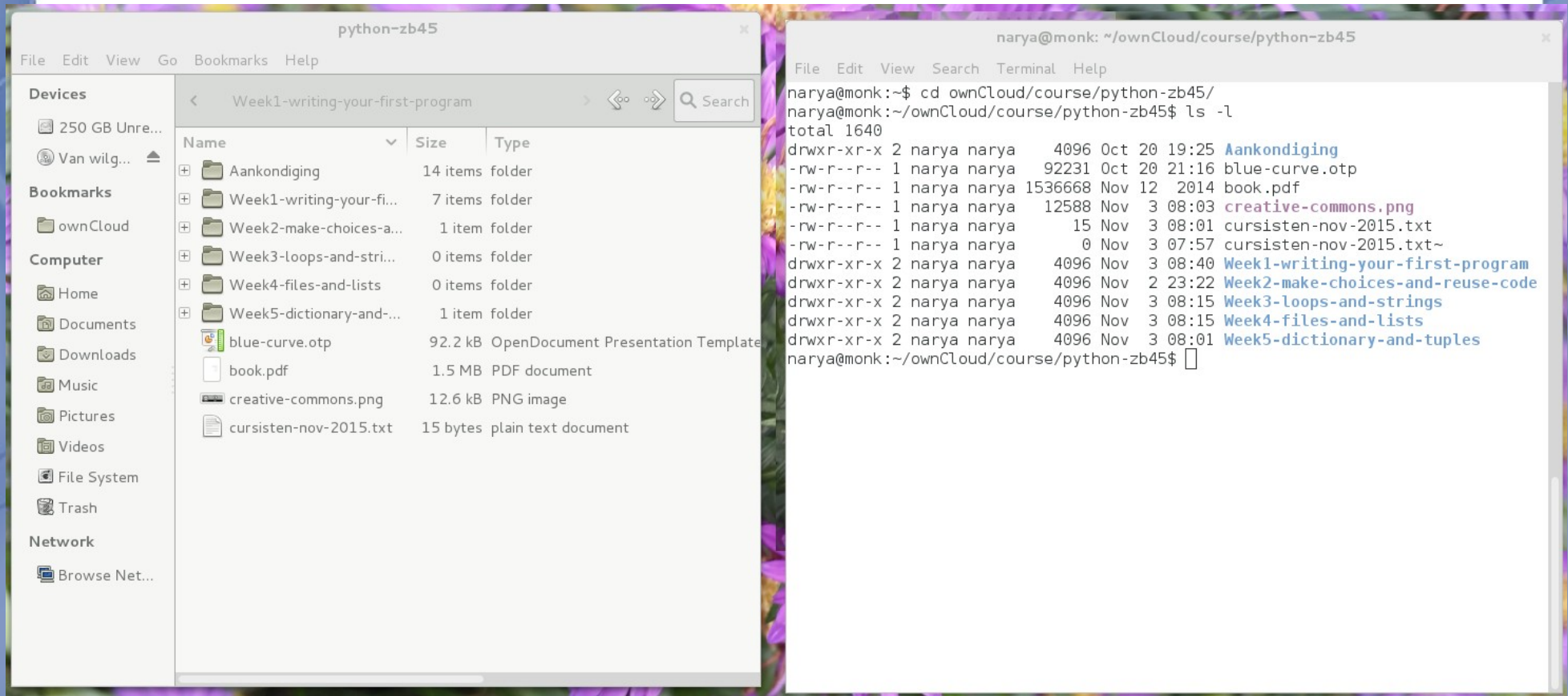
No icons

No clicking

You need to type in  
commands to the computer

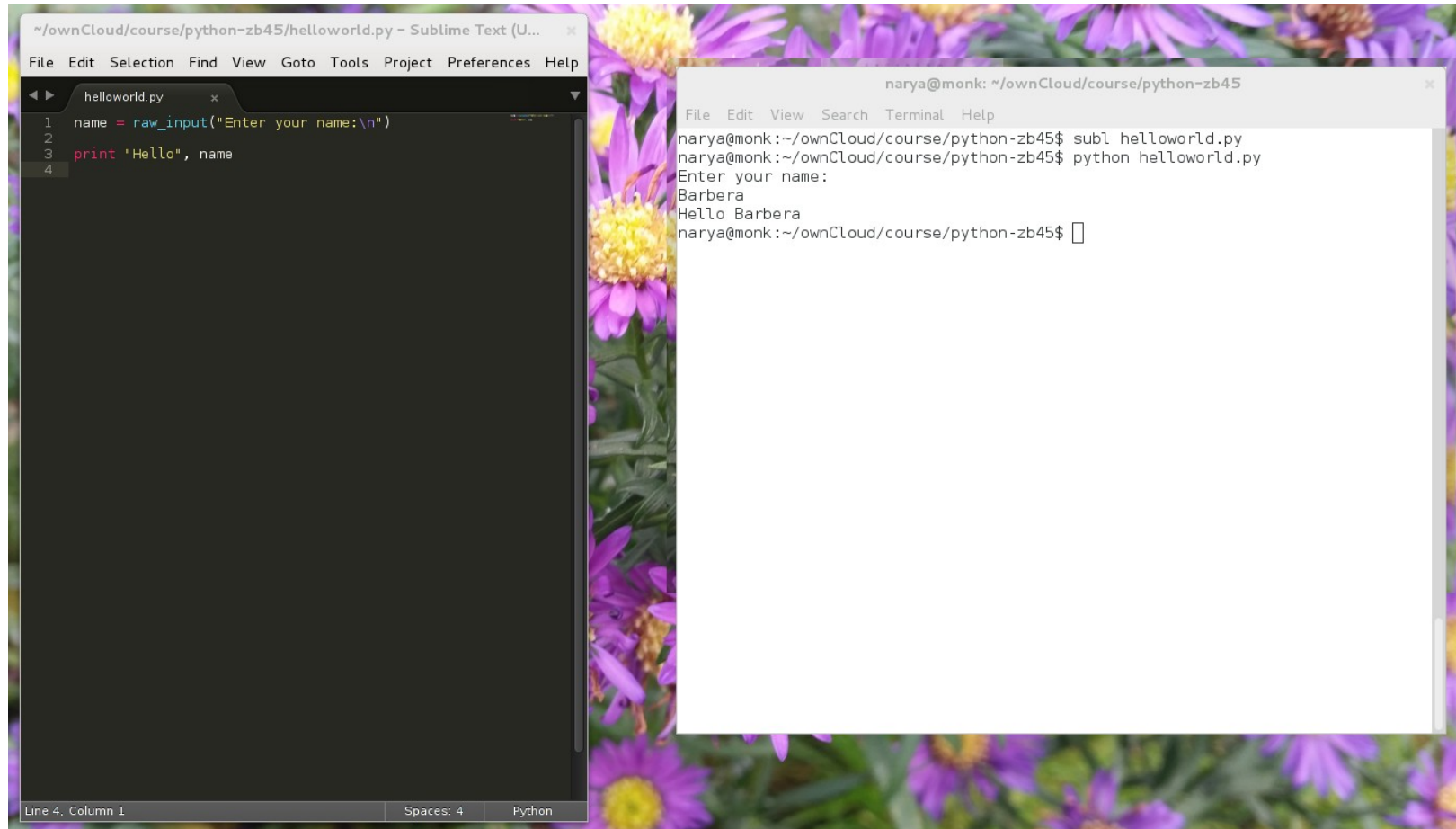


# Directories and files



`cd somedirectoryname` (go to directory, or Change Directory)  
`dir` (shows content of directory, windows)  
`ls -l` (shows content of directory, linux/mac)

# Starting a program



The image shows two overlapping windows against a background of purple flowers. On the left is a Sublime Text editor window titled '~/.ownCloud/course/python-zb45/helloworld.py - Sublime Text (U...'. It contains a Python script named 'helloworld.py' with the following code:

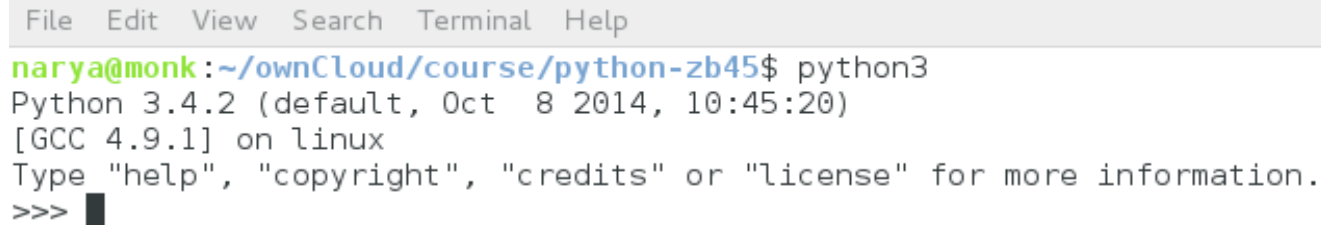
```
1 name = raw_input("Enter your name:\n")
2
3 print "Hello", name
4
```

On the right is a terminal window titled 'narya@monk: ~/.ownCloud/course/python-zb45'. It shows the following commands and output:

```
narya@monk:~/.ownCloud/course/python-zb45$ subl helloworld.py
narya@monk:~/.ownCloud/course/python-zb45$ python helloworld.py
Enter your name:
Barbera
Hello Barbera
narya@monk:~/.ownCloud/course/python-zb45$
```

The program “subl” is started (a text editor)  
After that the program “helloworld.py”  
Linux/mac: start “helloworld.py” with “python” before that  
Windows: you only need to type “helloworld.py”

# Start the Python program



```
File Edit View Search Terminal Help
narya@monk:~/ownCloud/course/python-zb45$ python3
Python 3.4.2 (default, Oct 8 2014, 10:45:20)
[GCC 4.9.1] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

You can also start the interactive python program

Linux/Mac: type “python”

Windows: start “python commandline” from the start menu

Note that the “prompt” has changed from blahblah\$ to >>>

You are now in the python program

You can type python code, one line at the time

# Python: first attempt

- Open python and type:

```
>>> Hi there
```

- Then type:

```
>>> answer me!
```

- One more try:

```
>>> help
```



# Python: help()

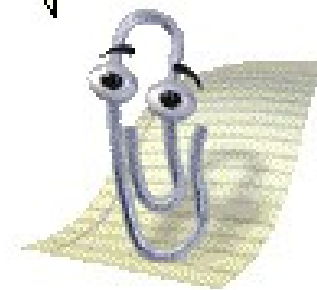
```
help> keywords
```

```
help> if
```

```
help> quit
```

It looks like you are bashing furiously on your keyboard. Do you want me to enable caps lock?

- ☒ Yes
- ☐ Die in a fire, Mr Clippy





# Hello world

```
>>> print("Hello world")
```

Try this again without the last quotes

```
>>> 27+15
```

```
>>> x = 7
```

```
>>> print(x)
```

```
>>> x = x+2
```

```
>>> print(x)
```

```
>>> 1+2=3
```

```
>>> 1+2==3
```



# Scripts

- Like a recipe, a set of instructions
- Open sublime, notepad++ or another editor
- Type the following in the editor and save the file as “myscript.py”
- Do not use spaces in the name of the script

```
x = 7
```

```
print(x)
```

```
x = x + 2
```

```
print(x)
```

Start the script from the commandline:

```
cmd> python myscript.py
```

Windows (if the above doesn't work):

```
cmd> myscript.py
```

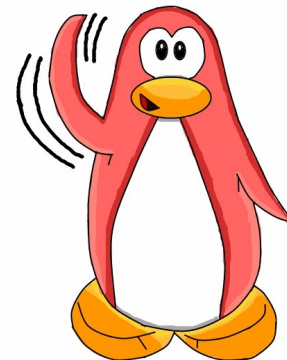
# Hello <your-name-here>

- New script “helloworld.py”

```
name = input("Enter your name:\n")  
print("Hello", name)
```

- Start the script from the commandline

```
cmd> python helloworld.py
```



# Data types

- Start an interactive Python session
- The basic data types:

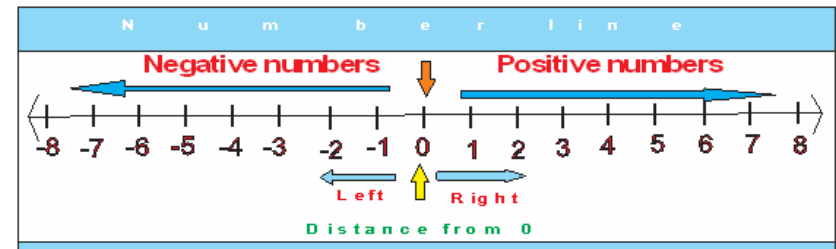
```
>>> type(4)
```

```
>>> type(3.141592653)
```

```
>>> type("Yo")
```

```
>>> type('Yo')
```

```
>>> type("13")
```



integer



float



string

# Variables

- Assign values to variables

```
>>> message = "hello"
```

```
>>> n = 2
```

```
>>> pi = 3.1415926535897931
```

```
>>> print(n)
```

```
>>> print(message)
```

```
>>> type(message)
```

```
>>> type(n)
```

```
>>> type(pi)
```





# Variable names

- A combination of letters, numbers and underscores
- No reserved words!
- No special characters!
- Case sensitive!
- Name has to start with letter or underscore
- Tip: use meaningful names



# Variable names

- Doesn't work:

```
>>> 7of9 = "borg"
```

```
>>> mail@ = "barbera@van-schaik.org"
```

- Works:

```
>>> city = "Amsterdam"
```

```
>>> pin_code = 1234
```



# Calculator

- Operators: +, -, \*, /, \*\*, %

```
>>> hour = 5
```

```
>>> minute = 59
```

```
>>> 20+32
```

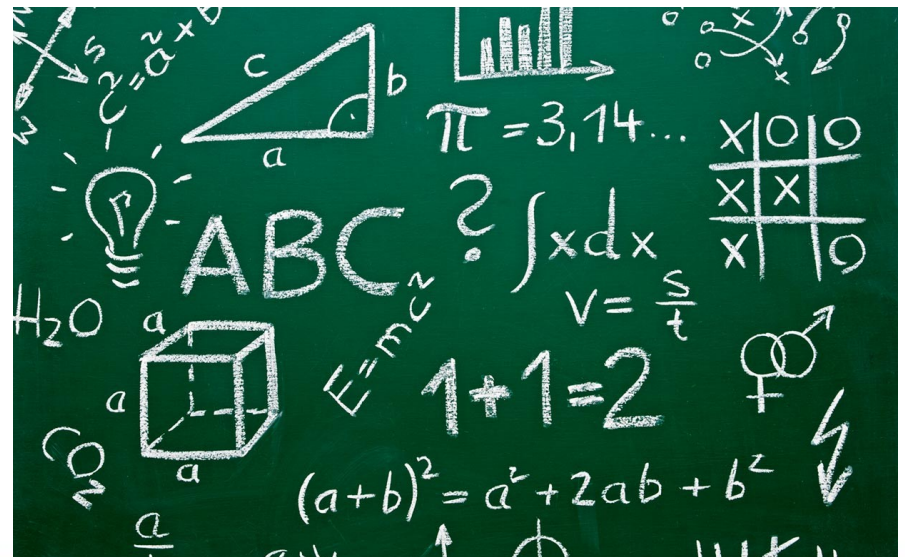
```
>>> hour - 1
```

```
>>> hour * 60 + minute
```

```
>>> minute / 60
```

```
>>> 5**2
```

```
>>> (5+9)*(15-7)
```



# Python 2 and floats

```
>>> minute = 59
```

```
>>> minute/60
```

```
>>> minute/60.0
```

In Python 3

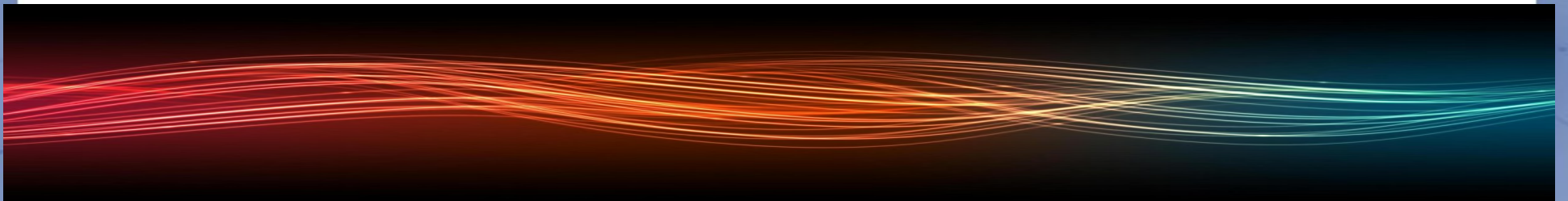
```
>>> minute//60
```

does same as python 2



# Order of operations

- Parenthesis:  $(1+2)*(3+1) = 3*4 = 12$
- Exponentiation:  $2**1+1 = 2+1 = 3$
- Multiplication and division:  $2*3-1 = 6-1 = 5$
- Addition and subtraction:  $5-3-1 = 2-1 = 1$
- Same precedence from left to right





# String operations

- Concatenation

```
>>> first = 10  
>>> second = 15  
>>> print(first + second)
```

```
>>> first = '10'  
>>> second = '15'  
>>> print(first + second)
```

- Multiplication

```
>>> word = "bla"  
>>> n = 7  
>>> print(n * word)
```



# Comments

blah, blah,  
blah, whatever

- Programs tend to get long and unreadable
- Solutions:
  - Give meaningful variable names
  - Insert comments

```
# Ask user for input
```

```
hours = float(input("How many hours did you work?\n"))
```

```
seconds = hours / 3600    # python sees this as correct
```

# Next week

- Next: Make choices and reuse code
- Want more practice?
  - Exercises in chapter 1 and 2 of the book
- See you next week!!

